

DAM
Desarrollo de Aplicaciones Multiplataforma
2º Curso

AD
Acceso a Datos

UD 8
Programación de componentes
de acceso a datos

IES BALMIS
Dpto Informática
Curso 2019-2020
Versión 1 (09/2019)

4. EJERCICIOS – Empaquetado y reutilización

UD8Ejer401 - Libros

Crear un proyecto para obtener el componente **Libros**, una clase para almacenar la información de libros, es decir:

```
private int id;  
private String titulo;  
private String autor;
```

El path del componente será **com.dam.bibliotecah**

Se debe obtener el archivo **Libros.jar** para su reutilización y distribución.

UD8Ejer402 - LibrosApp

Crear un proyecto denominado **LibrosApp** que reutilice el componente **Libros.jar** y ejecute las siguientes instrucciones:

```
Libros libro = new Libros(1, "Macbeth", "William Shakespeare");  
System.out.println(libro.toString());
```

UD8Ejer403 - PersonaSql

Crear un proyecto denominado **PersonaSql** para obtener un componente que almacene la información de la BD **bdpersonas** de MySQL.

El path del componente será **com.dam.persona**

```
CREATE SCHEMA bdpersonas DEFAULT CHARACTER SET utf8mb4 ;
```

```
CREATE TABLE personas (  
dni int(11) NOT NULL,  
nombre varchar(30) DEFAULT NULL,  
apellidos varchar(50) DEFAULT NULL,  
telefono varchar(15) DEFAULT NULL,  
PRIMARY KEY (dni)  
);
```

```
CREATE TABLE domicilios (  
id int(11) NOT NULL AUTO_INCREMENT,  
idpersona int(11) NOT NULL,  
direccion varchar(40) DEFAULT NULL,  
cpostal varchar(5) DEFAULT NULL,  
poblacion varchar(40) DEFAULT NULL,  
provincia varchar(40) DEFAULT NULL,  
PRIMARY KEY (id),  
FOREIGN KEY (idpersona) REFERENCES personas(dni)  
);
```

```

INSERT INTO personas (dni, nombre, apellidos, telefono)
VALUES
    (21111222, 'Pedro', 'Pérez', '666111222'),
    (21444555, 'Sergio', 'López', '666444555'),
    (21555666, 'Ana', 'González', '666555666');

INSERT INTO domicilios (idpersona, direccion, cpostal,
                        poblacion, provincia)
VALUES
    (21111222, 'el campillo, 12', '03004', 'Alicante', 'Alicante'),
    (21444555, 'la trapería', '03660', 'Novelda', 'Alicante'),
    (21555666, 'el cauce, 14', '46002', 'Cullera', 'Valencia');

```

El componente tendrá 3 clases:

- **Domicilio**: almacena los datos de un domicilio
- **Persona**: almacena los datos de una persona con su domicilio
- **ListaPersonas**: almacena los datos de la tabla personas y sus domicilios

```

public class Domicilio implements Serializable {
    String direccion;
    String cpostal;
    String poblacion;
    String provincia;
    ...
}

public class Persona implements Serializable {
    int dni;
    String nombre;
    String apellidos;
    String telefono;
    Domicilio domicilio;
    ...
}

public class ListaPersonas implements Serializable {
    ArrayList<Persona> lista = new ArrayList<Persona>();
    ...
}

```

Añade el método **cargaSql** a la clase **ListaPersonas** para que lea de MySQL y cargue todos los datos en el ArrayList:

```

public void cargaSql() {
    try {
        lista.clear();

        // Cargar el driver de mysql
        Class.forName("com.mysql.jdbc.Driver");

        // Cadena de conexión a MySQL
        String conURL = "jdbc:mysql://localhost/bdpersonas";
        conURL += "?user=root&password=1234&useSSL=false&autoReconnect=true";
    }
}

```

```

// Obtener la conexión
Connection con = (Connection) DriverManager.getConnection(conURL);

Statement stmt = (Statement) con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * "+
    "FROM personas, domicilios "+
    "WHERE personas.dni=domicilios.idpersona");
while (rs.next()) {
    Domicilio dom = new Domicilio(rs.getString("direccion"),
        rs.getString("cpostal"), rs.getString("poblacion"),
        rs.getString("provincia"));
    lista.add(new Persona(rs.getInt("dni"),
        rs.getString("nombre"), rs.getString("apellidos"),
        rs.getString("telefono"), dom));
}

// Cerrar la conexión
con.close();

} catch (SQLException e) {
    System.out.println("SQL Exception: " + e.toString());
} catch (ClassNotFoundException cE) {
    System.out.println("Exception: " + cE.toString());
}
}

```

UD8Ejer404 - PersonaAppSql

Crea un proyecto denominado **PersonaAppSql** que reutilice el componente **PersonaSql.jar** y ejecute las siguientes instrucciones:

- Crear un objeto de ListaPersonas
- Cargar los datos de MySQL en el objeto
- Mostrar los datos por pantalla

```

ListaPersonas lista = new ListaPersonas();

lista.cargaSql();

for (int i = 0; i<lista.getList().size(); i++) {
    System.out.println(lista.getList().get(i).toString());
}

```

5. EJERCICIOS - Maven

UD8Ejer501 - MavenJson

Crea un proyecto denominado **MavenJson** con el contenido del proyecto **UD2Json01** de la unidad didáctica de Ficheros pero con Maven.

Se debe crear con el Id del Grupo a **com.dam**

Las dependencias a incorporar para disponer del JAR de **javax.json** son:

- javax.json : javax.json-api
 - 1.1.4 [jar] – central
- org.glassfish : javax.json
 - 1.1.4 [jar] – central

UD8Ejer502 - MavenMySQL

Crea un proyecto denominado **MavenMySQL** con el contenido del proyecto **ConsultaBD1** de la unidad didáctica de "Bases de Datos Relacionales" pero con Maven.

Se debe crear con el Id del Grupo a **com.dam**

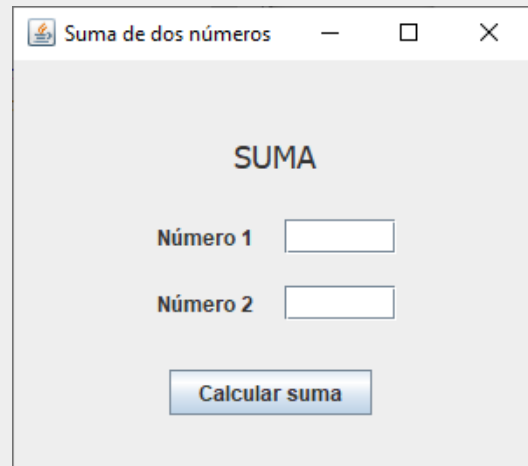
Las dependencias a incorporar para disponer del JAR de JDBC para conectar con MySQL **mysql-connector-java** son:

- mysql : mysql-connector-java
 - 5.1.48 [jar] – central

6. EJERCICIOS - Swing

UD8Ejer601 - SwingSuma

Utilizando Swing, crea una aplicación de escritorio de tipo Window con un JFrame que pida dos números y al pulsar un botón muestre en una ventana de mensajes la suma.



La clase de la Ventana se llamará WinFrame y será de tipo JFrame, deberá tener:

- JLabel: para las etiquetas "Número 1" y para "Número 2"
- JTextField: para poder introducir los dos números enteros
- JButton: para el botón que realizará la suma

El botón realizará el cálculo:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        JOptionPane.showMessageDialog(null,
            "La suma es " +
                (Integer.valueOf(this.jTextField1.getText())+
                 Integer.valueOf(this.jTextField2.getText())));
    } catch (NumberFormatException nfe) {
        JOptionPane.showMessageDialog(null,
            "Compruebe que los dos números son enteros");
    }
}
```

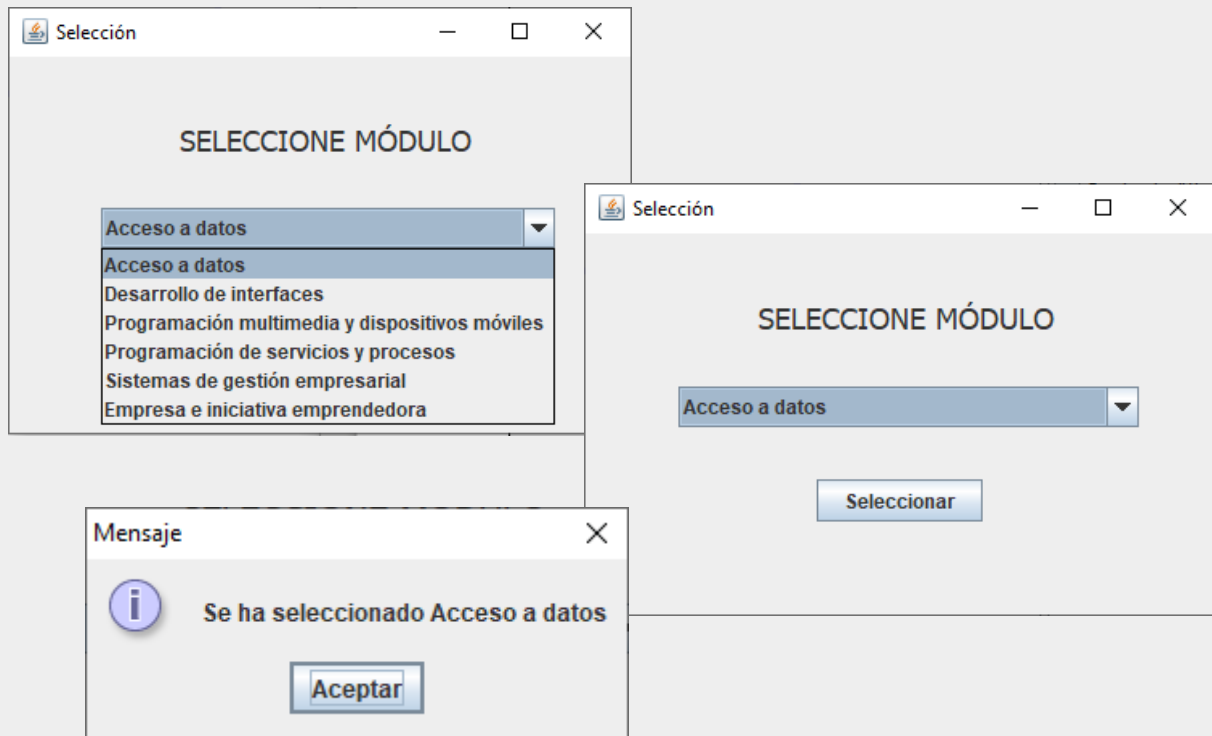
El main activará ventana:

```
public class SwingSuma {
    private static JFrame ventana;

    public static void main(String[] args) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                ventana = new WinJFrame();
                ventana.setLocationRelativeTo(null); // Para centrarla
                ventana.setTitle("Suma de dos números");
                ventana.setVisible(true);
            }
        });
    }
}
```

UD8Ejer602 - SwingDesplegable

Utilizando Swing, crea una aplicación de escritorio de tipo Window en la que el usuario seleccione de un desplegable un módulo de 2º de DAM y pulsando el botón se muestre el módulo seleccionado.



En la clase WinJFrame se deberá añadir un nuevo constructor que pueda recibir el ArrayList con el contenido a desplegar y lo incorpore al desplegable.

```
public WinJFrame(ArrayList<String> lista) {
    initComponents();
    for (int i=0; i<lista.size(); i++) {
        this.jComboBox1.addItem(lista.get(i));
    }
}
```

Y al pulsar el botón se deberá mostrar el módulo seleccionado:

```
JOptionPane.showMessageDialog(null,
    "Se ha seleccionado " + jComboBox1.getSelectedItem());
```

En la clase principal, el método `main` deberá crear la ventana:

```
public static void main(String[] args) {

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            ArrayList<String> modulos = new ArrayList<>();
            modulos.add("Acceso a datos");
            modulos.add("Desarrollo de interfaces");
            modulos.add("Programación multimedia y dispositivos móviles");
            modulos.add("Programación de servicios y procesos");
            modulos.add("Sistemas de gestión empresarial");
            modulos.add("Empresa e iniciativa emprendedora");

            ventana = new WinJFrame(modulos);
            ventana.setLocationRelativeTo(null); // Para centrarla
            ventana.setTitle("Selección");

            ventana.setVisible(true);
        }
    });
}
```

UD8Ejer603 - SwingDirChooser

Crea un proyecto denominado **SwingDirChooser** que permita al usuario elegir una carpeta y la muestre por consola.

Existen varias formas de averiguar carpetas del sistema, del usuario y de la aplicación.

```
public static void main(String[] args) {

    System.out.println("App User Directory :"+
        System.getProperty("user.dir") );
    System.out.println("Home User Directory :"+
        System.getProperty("user.home") );
    System.out.println("Home Directory :"+
        FileSystemView.getFileSystemView().getHomeDirectory() );
    System.out.println("Default Directory :"+
        FileSystemView.getFileSystemView().getDefaultDirectory() );
    System.out.println("Actual Directory :"+
        (new File(".").getAbsolutePath()) );

    JFileChooser jfc = new JFileChooser(new File(".").getAbsolutePath());
    jfc.setDialogTitle("Elegir una carpeta: ");
    jfc.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);

    int returnValue = jfc.showSaveDialog(null);
    if (returnValue == JFileChooser.APPROVE_OPTION) {
        if (jfc.getSelectedFile().isDirectory()) {
            System.out.println("Ha seleccionado la carpeta: " +
                jfc.getSelectedFile());
        }
    }
}
```


UD8Ejer604 - SwingTecladoMessage

Crea un proyecto denominado **SwingTecladoMessage** que pida al usuario teclear un String y muestre una ventana de tipo mensaje.

```
public static void main(String[] args) {  
  
    String nombre = JOptionPane.showInputDialog("Introduce tu nombre");  
  
    JOptionPane.showMessageDialog(null, "Cuidado "+nombre,  
                                "AVISO", JOptionPane.WARNING_MESSAGE );  
  
    System.out.println(nombre);  
}
```